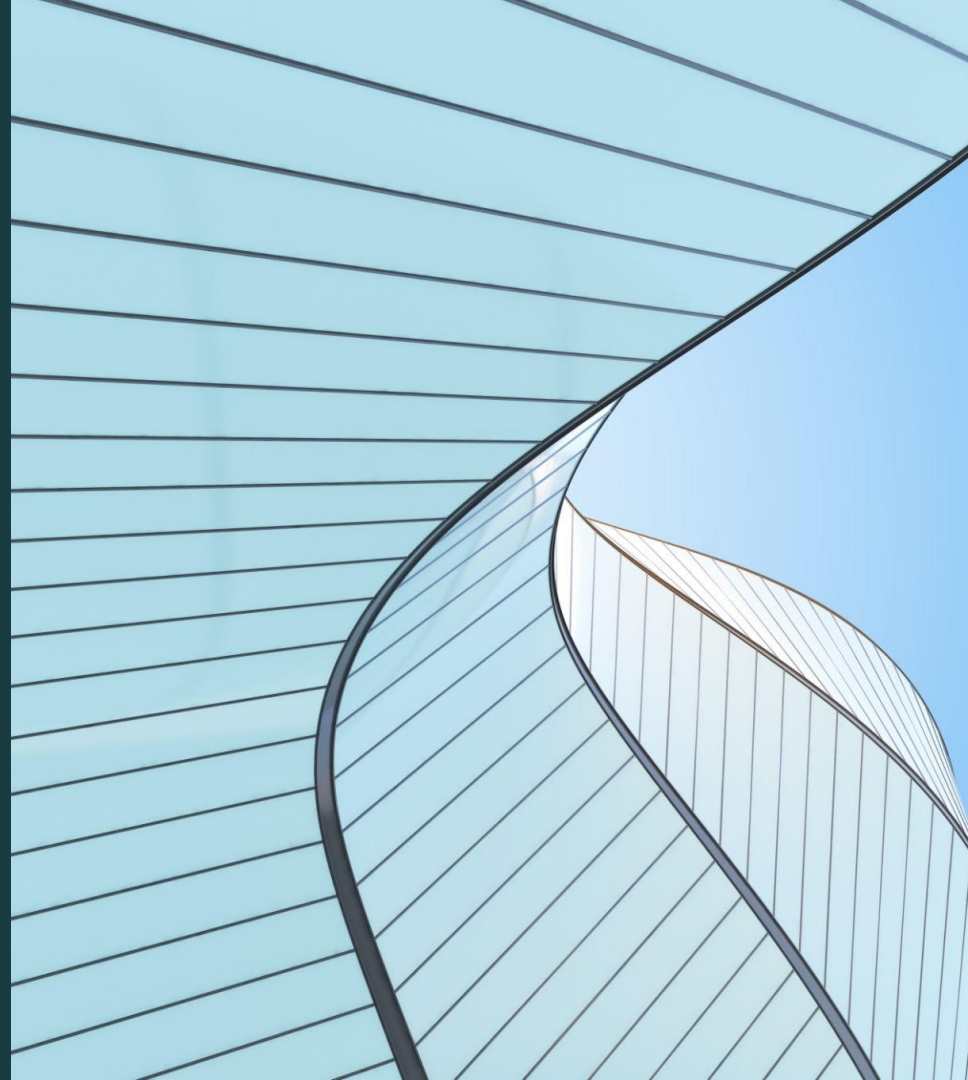


CMPE-297 Final Project: “PokeTrade Shopkeeper” Agent

By: Chint Patel, Dustin Nguyen, Nikhil Khanchandani ,
Vikranth Jakamukala



Content *overview*

1

Abstract

2

Objective

3

Overview

4

Approach

5

Architecture

6

Software Used

7

Future Work

Abstract

- This project demonstrates the design and deployment of an agentic AI application that simulates a digital shop with an intelligent shopkeeper. The agent uses a large language model to interact naturally with users, negotiate prices, and follow defined business rules. The system shows how modern AI agents can combine reasoning, personality, and tool usage within a cloud-deployed application.



Objective

- Build an autonomous AI agent capable of interactive decision-making
- Implement goal-driven conversational behavior instead of simple chat responses
- Integrate a large language model with an agent framework
- Deploy a full end-to-end agentic system to the cloud
- Demonstrate real-world use cases of agentic AI in commerce scenarios

Overview: PokeTrade Shopkeeper

- The application represents a virtual Pokémon card trading shop where:
 - Users interact with an AI shopkeeper through a web interface
 - The shopkeeper responds intelligently to bargaining attempts for Pokémon cards
 - The agent follows predefined trading rules while adapting responses dynamically
 - All reasoning is handled by a cloud-hosted AI model
- This project focuses on agent behavior, autonomy, and deployment, rather than static prompt-based chat.

Approach

- Agent Design
 - Defined a domain-specific shopkeeper persona for Pokémon card trading
 - Configured a confident, charismatic sales personality optimized for negotiation
 - Encoded pricing logic and behavioral constraints through structured instructions
- Model Integration
 - Integrated a large language model to handle reasoning and dialogue generation
 - Used instruction-based control to ensure consistent trading behavior
- Agent Framework Setup
 - Implemented the agent using an agent development framework
 - Enabled structured execution flow and state management
- Frontend Interaction
 - Built a web-based interface for real-time interaction with the agent
 - Supported conversational bargaining and product inquiries
- Cloud Deployment
 - Deployed the agentic application as a cloud-hosted service
 - Ensured scalable, remote access for user interaction

Architecture

- Web UI for user interaction
- Backend agent logic handling reasoning and decisions
- Large language model for natural language understanding
- Cloud hosting platform for deployment and execution
- The architecture follows a modular agentic design, making it easy to extend or add new agents later.



Software Used

- Large Language Model: Gemini (for reasoning and conversation)
- Agent Framework: Google Agent Development Kit (ADK)
- Backend Language: Python
- Cloud Platform: Google Cloud (Cloud Run)
- Frontend: Web-based UI (HTML/JavaScript)
- Development Environment: Cloud Shell / Local Terminal

Future Work

- Adding multiple agents (e.g., manager, accountant, delivery agent)
Introducing memory to track past users and preferences
 - Connecting real inventory databases
 - Implementing adaptive pricing strategies
- Improving UI/UX for mobile and accessibility
- Expanding to multi-agent collaboration scenarios

DEMO

Thank you